Task_1_prodigy_infotech_internship

June 4, 2024

PRODIGY INFOTECH DATA SCIENCE INTERN

#TASK 1

TASK OVERVIEW: Create a bar chart or histogram to visualize the distribution of a categorical or continuous variable, such as the distribution of ages or genders in a population.

[ ]:

```
#Here import the necessary libraries for this task

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Importing the population dataset here.

[ ]:

```
df1 = pd.read_csv("/content/Population data.csv", encoding="latin-1")  #␣
 ↪Replace with the correct encoding
```

Here I have checked about the dataset and it's statistical overview.

[ ]:

```
df1.head()
```

[ ]:

```
   rank  finalWorth               category                personName   age  \
0     1      211000       Fashion & Retail  Bernard Arnault & family  74.0
1     2      180000             Automotive                 Elon Musk  51.0
2     3      114000             Technology                Jeff Bezos  59.0
3     4      107000             Technology             Larry Ellison  78.0
4     5      106000  Finance & Investments            Warren Buffett  92.0

         country    city              source             industries  \
0         France   Paris                LVMH       Fashion & Retail
1  United States  Austin       Tesla, SpaceX             Automotive
2  United States  Medina              Amazon             Technology
3  United States   Lanai              Oracle             Technology
4  United States   Omaha  Berkshire Hathaway  Finance & Investments

  countryOfCitizenship  … cpi_change_country           gdp_country  \
0               France  …                1.1    $2,715,518,274,227
1        United States  …                7.5   $21,427,700,000,000
2        United States  …                7.5   $21,427,700,000,000
```

```
3        United States  …                7.5  $21,427,700,000,000
4        United States  …                7.5  $21,427,700,000,000

  gross_tertiary_education_enrollment  \
0                                65.6
1                                88.2
2                                88.2
3                                88.2
4                                88.2

  gross_primary_education_enrollment_country life_expectancy_country  \
0                                      102.5                    82.5
1                                      101.8                    78.5
2                                      101.8                    78.5
3                                      101.8                    78.5
4                                      101.8                    78.5

  tax_revenue_country_country total_tax_rate_country population_country  \
0                        24.2                   60.7         67059887.0
1                         9.6                   36.6        328239523.0
2                         9.6                   36.6        328239523.0
3                         9.6                   36.6        328239523.0
4                         9.6                   36.6        328239523.0

  latitude_country longitude_country
0        46.227638          2.213749
1        37.090240        -95.712891
2        37.090240        -95.712891
3        37.090240        -95.712891
4        37.090240        -95.712891

[5 rows x 35 columns]
```

Separate our requires coloumns i.e. Gender and Age

[ ]:
```
df = df1[['age','gender']]
```

[ ]:
```
df
```

[ ]:
```
       age gender
0     74.0      M
1     51.0      M
2     59.0      M
3     78.0      M
4     92.0      M
…      …    …
2635  51.0      M
```

```
 2636  80.0      M
 2637  60.0      M
 2638  71.0      M
 2639  66.0      M

 [2640 rows x 2 columns]
```

```
[ ]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2640 entries, 0 to 2639
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   age     2575 non-null   float64
 1   gender  2640 non-null   object
dtypes: float64(1), object(1)
memory usage: 41.4+ KB
```

Handling the missing values

```
[ ]: df.isnull().sum()
```

```
[ ]: age       65
     gender     0
     dtype: int64
```

```
[ ]: median_age = df['age'].median()  # Calculate the median age
     df['age'] = df['age'].fillna(median_age)  # Replace missing values with median
```

```
<ipython-input-64-c1cdd41d1419>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df['age'] = df['age'].fillna(median_age)  # Replace missing values with median
```

```
[ ]: df['age'].isnull().sum()
```

```
[ ]: 0
```

Our task is to visualize continuous or categorical variables such as Gender and Age through Bar Chart and Histogram.

Distribution of Gender

```
[ ]: Gender_count = df['gender'].value_counts()
     Gender_count
```

[ ]:

```
gender
M    2303
F     337
Name: count, dtype: int64
```

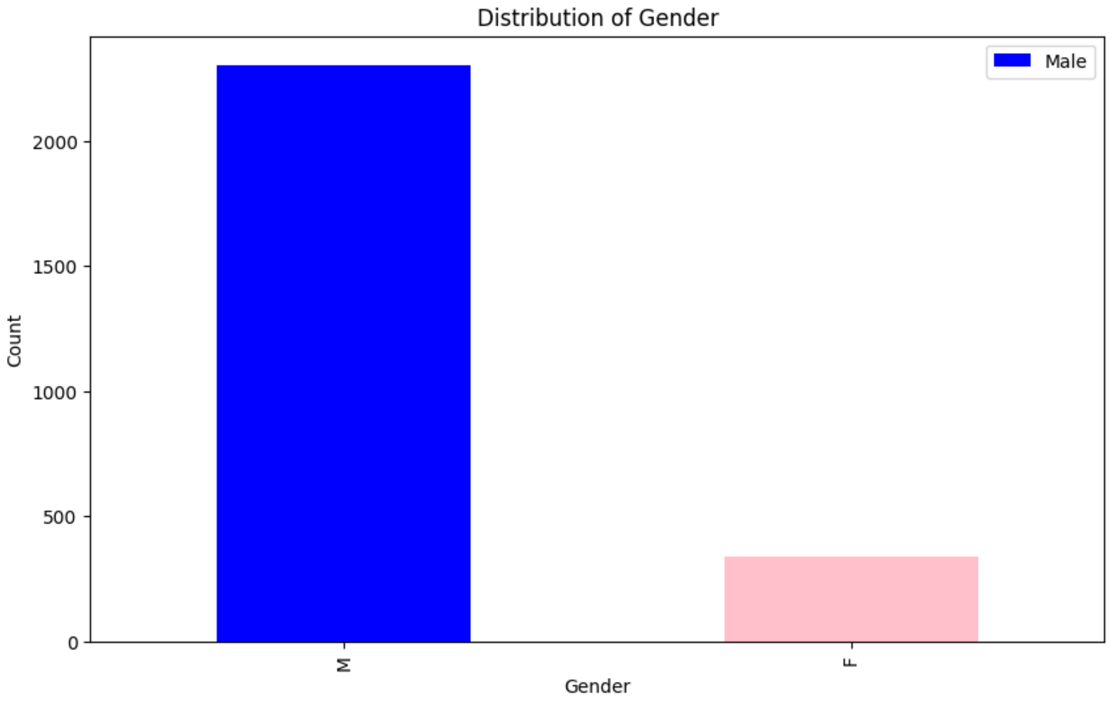
[ ]:

```python
plt.figure(figsize=(10, 6))  # Set the figure size

# Plot the bar chart with colors
Gender_count.plot(kind="bar", color=["Blue", "Pink"])

# Add the legend at the top right corner
plt.legend(labels=["Male",
                   "Female"],loc="upper right")


plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Distribution of Gender')
plt.show()
```



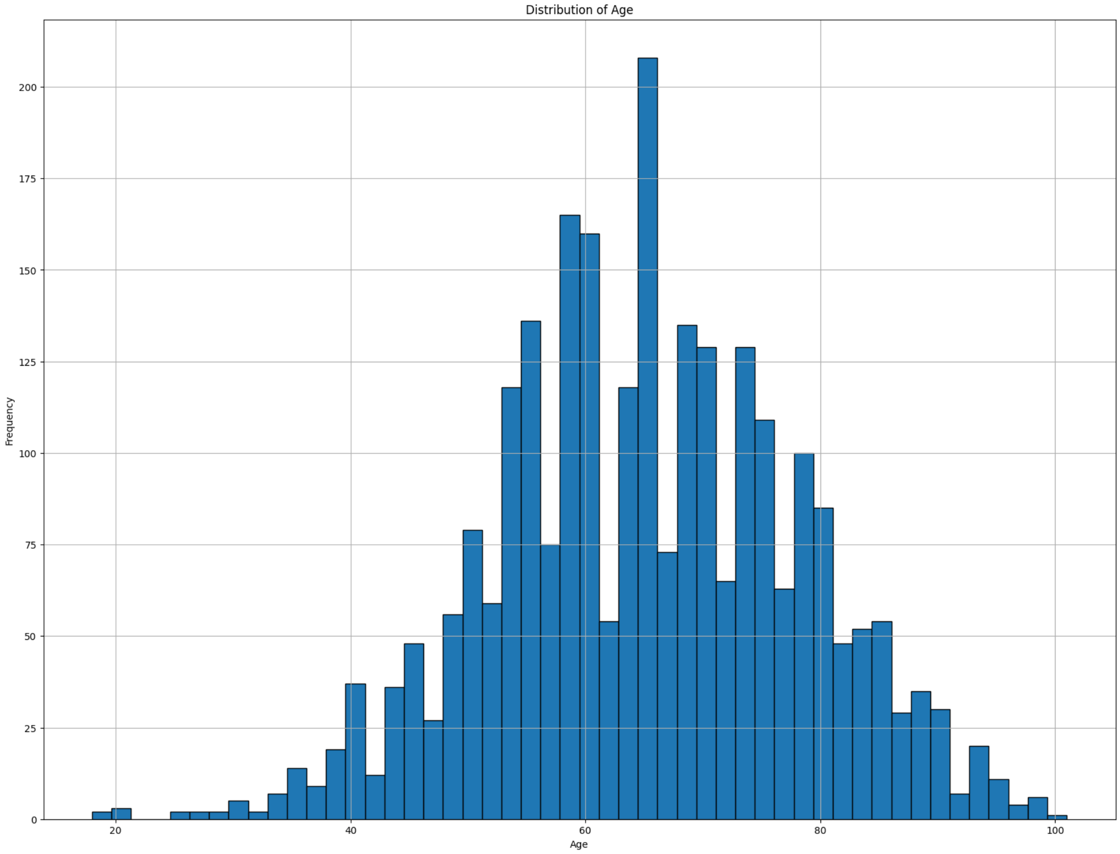
Distribution of Age

[ ]:

```python
df['age'].describe()
```

```
[ ]: count    2640.000000
     mean       65.136742
     std        13.093821
     min        18.000000
     25%        56.000000
     50%        65.000000
     75%        74.000000
     max       101.000000
     Name: age, dtype: float64
```

```
[ ]: df['age'].hist(bins=50,figsize=(20,15),edgecolor='black')
     plt.xlabel('Age')
     plt.ylabel('Frequency')
     plt.title('Distribution of Age')
     plt.show()
```


Distribution of Age
Frequency
Age